



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/650,245	08/27/2003	Wolfgang Grieskamp	3382-64897	7189
26119	7590	07/30/2008	EXAMINER	
KLARQUIST SPARKMAN LLP			DAO, THUY CHAN	
121 S.W. SALMON STREET				
SUITE 1600			ART UNIT	PAPER NUMBER
PORLTAND, OR 97204			2192	
			MAIL DATE	DELIVERY MODE
			07/30/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)
	10/650,245	GRIESKAMP ET AL.
	Examiner	Art Unit
	Thuy Dao	2192

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 03 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) Responsive to communication(s) filed on 16 April 2008.
 2a) This action is **FINAL**. 2b) This action is non-final.
 3) Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) Claim(s) 1,3-8,10,11,17,19 and 30-37 is/are pending in the application.
 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
 5) Claim(s) _____ is/are allowed.
 6) Claim(s) 1,3-8,10,11,17,19 and 30-37 is/are rejected.
 7) Claim(s) _____ is/are objected to.
 8) Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) The specification is objected to by the Examiner.
 10) The drawing(s) filed on 27 August 2003 is/are: a) accepted or b) objected to by the Examiner.
 Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
 Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
 11) The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
 a) All b) Some * c) None of:
 1. Certified copies of the priority documents have been received.
 2. Certified copies of the priority documents have been received in Application No. _____.
 3. Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|--|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413) |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | Paper No(s)/Mail Date. _____ . |
| 3) <input checked="" type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date <u>12/06/06 (resubmitted 04/16/08)</u> . | 5) <input type="checkbox"/> Notice of Informal Patent Application |
| | 6) <input type="checkbox"/> Other: _____ . |

DETAILED ACTION

1. This action is responsive to the amendment filed on April 16, 2008.
2. Claims 1, 3-8, 10, 11, 17, 19, and 30-37 have been examined.

Response to Amendments

3. In the instant amendments, claims 2, 8, 19, 35-37 have been amended.
4. The objection to the claims is withdrawn in view of Applicants' amendments.

Information Disclosure Statement

5. The Office acknowledges receipt of the Information Disclosure Statement filed on December 6, 2006 and resubmitted on April 16, 2008. It has been placed in the application file and the information referred to therein has been considered by the examiner.

Specification

6. In the last Office action mailed February 5, 2008, the amendments (page 2) to the specification were not entered because the corrections at different locations all direct to the same page 6, line 13 of the specification.

In this Office action, the amendments (pages 2-3) to the specification has not been entered because the last correction directs to a wrong location in the originally filed disclosure.

The examiner respectfully requests the Applicants resubmit the amendments to the specification with a correct line number - -Please replace the paragraph beginning at page 25, line [[29]] 19, with the following rewritten paragraph: ...- -.

Appropriate correction is required.

Claim Objections

7. Claim 3 is objected to because of minor informality. The phrase in line 1 is considered to read as - -The computer readable medium of claim [[3]] 1 further comprising ...- -.

Appropriate correction is requested.

Response to Arguments

8. Applicants' arguments have been considered but are moot in view of the new ground(s) of rejection.

Claim Rejections – 35 USC § 101

9. 35 U.S.C. 101 reads as follows:

Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

10. Claims 1, 3-6, 17, and 30-31 are directed to a computer readable medium, which may include communication medium (specification, page 26, line 8).

A computer readable medium product is a tangible physical article or object, some form of matter, which a signal is not. That the other two product classes, machine and composition of matter, require physical matter is evidence that a manufacture was also intended to require physical matter. A signal, a form of energy, does not fall within either of the two definitions of manufacture. Thus, a signal does not fall within one of the four statutory classes of Sec. 101 – see MPEP 2106

Under the principles of compact prosecution, claims 1, 3-6, 17, and 30-31 have been examined as the Examiner anticipates the claims will be amended to obviate these 35 USC § 101 issues. For example (constructive suggestion only), - -A/The computer readable storage medium ... - as disclosed in the specification, page 25: 26 - page 26: 5.

Claim Rejections – 35 USC § 103

11. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious

at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

12. Claims 1, 3-8, 10, 11, 17, 19, and 30-37 are rejected under 35 U.S.C. 103(a) as being unpatentable over US Patent No. 5,937,186 to Horiguchi et al. (art made of record, hereafter "Horiguchi") in view of US Patent No. 6,708,194 to Korn (art made of record, hereafter "Korn").

Claim 1:

Horiguchi discloses a *computer readable storage medium having computer-executable software code carried thereon for executing on a computing device for saving state for a semantically accessible state binding method, comprising:*

a data structure comprising: a first state frame including a representation of a first state of an executing program (e.g., FIG. 5, col.8: 5-32, first stack frame 504 including first state/context of an executing program such as context info of the calling program, dynamic variables; col.7: 9-53), and

a second state frame including a representation of a second state of the executing program comprising state changes made by the executing program after the first state frame is created (e.g., FIG. 5, col.8: 5-32, second stack frame 506) and

the second state frame includes a pointer back to the first state frame (e.g., FIG. 5, in second stack frame 506, Backward Pointer 408 pointing back to first stack frame 504);

a third state frame including a representation of state changes made by the executing program, and the third state frame includes a pointer back to the second frame (e.g., FIG. 5, col.8: 5-32, third stack frame 512);

wherein the executing program saves a first state of the program in the first state frame as a semantically accessible first state object (e.g., FIG. 5, first stack frame 504 has BckPtr 408 containing the entry address of said first stack frame 504 as the root address of the stack),

saves a second state of the program in the second state frame as a semantically accessible second state object (e.g., FIG. 5, second stack frame 506 has BckPtr 408 as a backward pointer containing the entry address of first stack frame by using an assignment instruction such as “second_frame.BckPtr = address(first_frame.BckPtr)”, i.e., saving address of the entry address of the first frame to “second_frame.BckPtr” for backward traversing later when popping the stack), and

then returns to the first state of the program by using the first state object (e.g., FIG. 5, stack POINTER 510, wherein the executing program returns to the first stack frame 504 from the second stack frame 506 (popping the stack) by using the an assignment instruction such as “POINTER = second_frame.BckPtr”).

Horiguchi does not explicitly disclose other limitations. However, in an analogous art, Korn further discloses:

a third state frame including a representation of state changes made by the executing program after a fork method creates the third state frame (e.g., col.11: 10-48; FIG. 7, col.14: 6-35), and

the third state frame includes a pointer back to the second frame (e.g., col.14: 22-35; col.15: 55 – col.16: 12);

wherein the third state frame is empty when created by the fork method (e.g., col.6: 33-54, generating a new (empty) frame in the forked (child) process/thread and after that, copying state/context of forking (parent) process/thread to said new frame; col.7: 63 – col.8: 9); and

wherein, after the fork method creates the third state frame, value of a variable of the executing program can be accessed by checking, in backwards order that the frames were created, for the value of the variable in the respective frames (e.g., col.6: 32-46; col.10: 54-63; col.11: 10-26; col.13: 62 – col.14: 35).

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to combine Korn's teaching into Horiguchi's teaching. One would have been motivated to do so to handle primary (parent) thread (parent) and

other (child) threads when parallel executing as suggested by Korn (e.g., col.3: 47 - col.4: 14).

Claim 3:

The rejection of claim 2 is incorporated. Horiguchi discloses *a fourth state frame which includes changes made by the executing program after the fork method creates the third state frame and after a set method returns the executing program to the state of the second state frame, and wherein the fourth state frame includes a pointer back to the second state frame* (e.g., FIG. 5, col.8: 5-32).

Claim 4:

The rejection of claim 3 is incorporated. Korn discloses *a joined state frame including a combination of state changes in the third and fourth state frames* (e.g., col.11: 10-48; col.14: 6-35).

Claim 5:

The rejection of claim 3 is incorporated. Horiguchi discloses *a first thread of the executing program makes state changes copied in the second state frame, and a second thread of the executing program makes state changes copied into the third state frame* (e.g., col.7: 63 – col.8: 9).

Claim 6:

The rejection of claim 1 is incorporated. Horiguchi discloses *the second state frame includes unchanged state read from the first state frame* (e.g., col.7: 9-53).

Claim 7:

Horiguchi discloses a *computerized method comprising:*

receiving via an application programming interface a request to create a state save (e.g., FIG. 5, col.8: 5-32, creating state/context/variable saves in each stack frame);

in response to the request, saving a first representation of a state of an executing program comprising copying state of the program required to return to the moment the state was saved as a first state frame (e.g., col.8: 5-32; col.9: 16-57);

creating a second state frame with a backward link to the first state frame as a current state frame (e.g., col.8: 33-61);

maintaining a second representation of subsequent state comprising changes made to the state of the executing program after the first representation in the current state frame (e.g., col.7: 9-53);

in response to a request for value of a variable after the request to create a state save, checking for the value of the variable in the first state frame (e.g., col.8: 5-32); and

changing the current state frame to the first state frame upon receiving a state set request at the application programming interface (e.g., col.8: 33-61; col.9: 16-57).

Horiguchi does not explicitly disclose *creating a blank state frame with a backward link to the first state frame as a current state frame.*

However, Korn further discloses:

creating a blank state frame with a backward link to the first state frame as a current state frame (e.g., col.6: 33-54, generating a new (empty) frame in the forked (child) process/thread and after that, copying state/context of forking (parent) process/thread to said new frame; col.7: 63 – col.8: 9); and

value of a variable of the executing program can be accessed by checking, in backwards order that the frames were created, for the value of the variable in the respective frames (e.g., col.6: 32-46; col.10: 54-63; col.11: 10-26; col.13: 62 – col.14: 35).

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to combine Korn's teaching into Horiguchi's teaching. One would have been motivated to do so to handle primary (parent) thread (parent) and

other (child) threads when parallel executing as suggested by Korn (e.g., col.3: 47 - col.4: 14).

Claim 8:

Horiguchi discloses a computer system comprising: memory and a central processing unit executing,

a program including executable instructions and an evolving present state (e.g., FIG. 5, col.8: 5-32);

a state component comprising a representation of the first state frame as a programmer-accessible named object (e.g., FIG. 4, col.7: 9-53),

a second state frame comprising a subsequent representation of state changes made by the program since the initial representation (e.g., col.9: 16-57),

the program including a method for returning the program state to the prior evolving present state using the state component (e.g., FIG. 9, col.12: 19-67); and

the program including a method for locating a value updated during the prior evolving present state and not present in the second state frame by following a back pointer (e.g., col.8: 5-32; col.11: 54 - col.12: 18)

from the second state frame to the first state frame, reading location value from the first state frame and storing the location value in the second state frame (e.g., col.8: 10-61; col.7: 9-53).

Horiguchi does not explicitly disclose a first state frame comprising an initial representation of a prior evolving present state of the program.

However, Korn further discloses:

a first state frame comprising an initial representation of a prior evolving present state of the program (e.g., col.6: 33-54, generating a new (empty) frame in the forked (child) process/thread and after that, copying state/context ,i.e., a prior evolving present state, of forking (parent) process/thread to said new frame; col.7: 63 – col.8: 9); and

value of a variable of the executing program can be accessed by checking, in backwards order that the frames were created, for the value of the variable in the respective frames (e.g., col.6: 32-46; col.10: 54-63; col.11: 10-26; col.13: 62 – col.14: 35).

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to combine Korn's teaching into Horiguchi's teaching. One would have been motivated to do so to handle primary (parent) thread (parent) and other (child) threads when parallel executing as suggested by Korn (e.g., col.3: 47 - col.4: 14).

Claim 10:

The rejection of claim 8 is incorporated. Korn discloses *the state component includes a fork method for maintaining state for a thread spawned by the program and a forked representation of state changes made by the spawned thread of the program* (e.g., col.6: 33-46; col.14: 22-35).

Claim 11:

The rejection of claim 10 is incorporated. Korn discloses *the state component includes a join method for joining state changes made by the forked thread back into state changes of the subsequent representation* (e.g., col.7: 63 – col.8: 9; col.15: 55 – col.16: 12).

Claim 17:

Claim 17 is a computer readable storage medium version, which recites the same limitations as those of claim 7, wherein all claimed limitations have been addressed and/or set forth above. Therefore, as the references teach all of the limitations of the above claim, they also teach all of the limitations of claim 17.

Claim 19:

Horiguchi discloses a computerized method comprising computer executable instructions for performing a method comprising:

receiving a request from a method, which takes as a parameter a state object, to create a saved state of an executing model (e.g., col.8: 5-32; col.9: 16-57);

saving a first representation of a state of the executing model as a first state frame; saving a first representation of the state frame as the state object (e.g., col.7: 9-53; col.12: 19-67);

maintaining, in the second state frame, a second representation of state changes made by the executing model after the first representation as the state changes occur (e.g., col.8: 33-61; col.13: 51-67); and

reinstating the executing model state to the state of the first representation using the state object (e.g., col.8: 5-32; col.11: 54 – col.12: 18).

Horiguchi does not explicitly disclose creating a blank state frame with a backward link to the first state frame as a second state frame.

However, Korn further discloses:

creating a blank state frame with a backward link to the first state frame as a second state frame (e.g., col.6: 33-54, generating a new (empty) frame in the forked (child) process/thread and after that, copying state/context ,i.e., a prior evolving present state, of forking (parent) process/thread to said new frame; col.7: 63 – col.8: 9); and

value of a variable of the executing program can be accessed by checking, in backwards order that the frames were created, for the value of the variable in the respective frames (e.g., col.6: 32-46; col.10: 54-63; col.11: 10-26; col.13: 62 – col.14: 35).

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to combine Korn's teaching into Horiguchi's teaching. One would have been motivated to do so to handle primary (parent) thread (parent) and other (child) threads when parallel executing as suggested by Korn (e.g., col.3: 47 - col.4: 14).

Claim 30:

The rejection of claim 3 is incorporated. Horiguchi discloses *the data structure further comprises a fifth state frame with a pointer to the fourth state frame, wherein the fourth state frame further comprises a reference pointer with a value indicating how many frames point back to it, and wherein if the value of the reference pointer is one, then the executing program combines the fourth state frame with the fifth state frame* (e.g., FIG. 9, col.12: 19-67).

Claim 31:

The rejection of claim 3 is incorporated. Horiguchi discloses *the fourth state frame includes a pointer to the third state frame* (e.g., col.8: 33-61).

Claim 32:

The rejection of claim 7 is incorporated. Horiguchi discloses *writing the value of the variable in a cache in the current state frame* (e.g., FIG. 4, col.7: 9-53).

Claim 33:

The rejection of claim 32 is incorporated. Horiguchi discloses *a threshold size wherein when the cache is greater than the threshold size, the cache is purged* (e.g., FIG. 10C, col.13: 51-67).

Claim 34:

The rejection of claim 32 is incorporated. Horiguchi discloses *a threshold size wherein when the cache is greater than the threshold size, the last used variable is overwritten* (e.g., FIG. 10A, col.11: 54 - col.12: 18).

Claim 35:

The rejection of claim 19 is incorporated. Korn discloses:
receiving a request from a fork method which saves state of the executing program as a first state frame when invoked with the state object; receiving a request

from a set method which sets the state of the executing program to the first state when invoked with the state object (e.g., col.11: 10-48; col.14: 6-35); and

receiving a request from a join method which joins the current state of the executing program with the first state when invoked with the state object (e.g., col.6: 33-46; col.14: 22-35).

Claim 36:

The rejection of claim 19 is incorporated. Horiguchi discloses *the first frame and the second frame have a creation order* (e.g., col.8: 10-16).

Claim 37:

The rejection of claim 36 is incorporated. Horiguchi discloses *when a variable is accessed by the executing model, checking the state frames in opposite order to their creation order until an update for the variable is found* (e.g., col.8: 10-32; col.9: 16-57).

Conclusion

13. Any inquiry concerning this communication should be directed to examiner Thuy Dao (Twee), whose telephone/fax numbers are (571) 272 8570 and (571) 273 8570, respectively. The examiner can normally be reached on every Tuesday, Thursday, and Friday from 6:00AM to 6:00PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam, can be reached at (571) 272 3695.

The fax phone number for the organization where this application or proceeding is assigned is (571) 273 8300.

Any inquiry of a general nature of relating to the status of this application or proceeding should be directed to the TC 2100 Group receptionist whose telephone number is (571) 272 2100.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR.

Art Unit: 2192

Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

/Thuy Dao/

Examiner, Art Unit 2192

/Tuan Q. Dam/

Supervisory Patent Examiner, Art Unit 2192